

## P a t e n t   c l a i m s

1. Arrangement for simplifying the design and implementation of mobile services in a communication system, especially a telecommunications system, said system comprising distributed hardware and software components which interact in order to provide services to one or more users,  
c h a r a c t e r i z e d by introducing in said system  
10 a mobility transparency, for thereby enabling facilitated application design including inter alia terminal or personal mobility.
2. Arrangement as claimed in claim 1,  
15 c h a r a c t e r i z e d i n that said mobility transparency is adapted to hide all the necessary mechanism and functions, for thereby supporting mobility to the application in question and to the application designer.
- 20 3. Arrangement as claims in claim 1 or 2,  
c h a r a c t e r i z e d i n that said mobility transparency is to be considered as a new transparency in addition to already existing transparency, for example in  
25 addition to the one defined by ODP (Open Distributed Processing) and adopted by TINA (Telecommunication Information Networking Architecture).
4. Arrangement as claimed in any of the claims 1-3,  
30 c h a r a c t e r i z e d i n that mobility transparency is introduced to the application designers at the requirement and functional specification phases, i.e. by integrating mobility function (M) totally into the infrastructure of platform.

005007-422460

S. P.

5. Arrangement as claimed in any of the preceding claims,

5 characterized in that computational objects are mapped to engineering objects (EO) so as to be none-visible in the computational model of the application in question.

6. Arrangement as claimed in any of the preceding claims,

10 characterized in that the mobility function constitutes the engineering object interceptor, i.e. being arranged at the boundary between the terminal domain and the telecom system domain.

15 7. Arrangement as claimed in any of the preceding claims,

20 characterized in that an engineering model is developed by letting each computational object (CO) be mapped to one or more Basic Engineering Objects (BEOs), interaction between computational objects (CO1, CO2) belonging to the same cluster being effected directly as method calls, and communications between computational objects (CO3, CO4) located in the telecom system domain and in different clusters being effected through a channel comprising stubs, binders and protocols.

8. Arrangement as claimed in any of the preceding claims,

30 characterized in that communication between user domain computation objects (CO2) and telecom system domain computation object (CO3) residing in different clusters will take place in a channel comprising

Sub 1

5533344460

an interceptor ( ) in addition to the usual objects, said interceptor being invisible for the application designer.

9. Arrangement as claimed in any of the preceding  
5 claims,

characterized in that from the computational model the application designer can decide whether an object belongs to the user domain or the telecom system domain, for on the basis of such a decision to generate the necessary objects, such as stubs, for all the application objects.

10. Arrangement as claimed in any of claims 1-3,  
characterized in that the arrangement  
15 comprises means for sending a message to a routing broker asking for specific server to perform a task, said broker locating the server and sending said request to said server, said mobility functions (M) thereby playing the role as the broker.

20 11. Arrangement as claimed in any of the claims 1-3 or 10,  
characterized in that said mobility function (M) comprises a cascade of two brokers.

25 12. Arrangement as claimed in any of the claims 1-3 or 10-11,  
characterized in that said mobility functions (M) in the form of two brokers allows for interaction between an object (CO1) belonging to a user domain and an object (CO2) belonging to a telecom system domain by letting said interactions enter said mobility functions (M) at one end and exit therefrom at the other end (Fig. 5).

SECRET  
SUB A1

13. Arrangement as claimed in any of the claims 1-3 or 10-12,  
c h a r a c t e r i z e d i n that both ends of the  
5 mobility functions (M) support both the same interface type containing an "invoke type" operation, for thereby allowing a request to be built and invoked dynamically by client objects.
- 10 14. Arrangement as claimed in claim 13,  
c h a r a c t e r i z e d i n that there is used an invoke operation, which invoke comprises an object name (or identifier), an operation name, and a parameter list for the invoked operation.
- 15 15. Arrangement as claimed in any of the claims 1-3, or 10-14,  
c h a r a c t e r i z e d i n that the mobility func-  
20 tions (M) are introduced in the system architecture as a functional layer between the application layer and the DPE layer, said layer separation allowing every layers to use services offered by other layers.
- 25 16. Arrangement as claimed in any of the claims 1-3, or 10-15,  
c h a r a c t e r i z e d i n that there is intro-  
duced an intermediary model called derived computational model in the transition from the computational model to the engineering model, said intermediary model being used  
30 to map all interactions traversing the boundary between the user domain and the telecom system domain to interac-  
tions with the mobility functions (M).
17. Arrangement as claimed claim 16,

665007-4432460

Sub 8A2  
characterized in that on the basis of the derived computational model, an engineering model can be elaborated and engineering objects, such as stubs, can be mechanically generated.

5

18. Arrangement as claimed in any of the claims 1-3, characterized in that said arrangement comprises proxy objects, a proxy acting on behalf of an entity in a transparent way, i.e. when interacting with a proxy an entity cannot tell whether it is dealing with the real counterpart or with its proxy.

19. Arrangement as claimed in claim 18, characterized in that the proxy is adapted to marshall the operation invocation into an invoke operation at the respective end of the mobility functions (M).

20. Arrangement as claimed in claim 19, characterized in that the proxy is introduced as a symmetrical constellation if interactions are to be in both directions.

Sub 8B3  
21. Arrangement as claimed in claim 20, characterized in that the necessary proxies are introduced and created according to the communications needs, alternatively dynamically at run-time.

22. Arrangement as claimed in any of the claims 1-3 or 18-21, characterized in that there is introduced an object type designated Dynamic Object (DO), all proxies being of the type DO, and that a proxy, i.e. an

663007-422460

object (instance) of type DO is initiated from an object template called Dynamic Object Implementation (DOI).

23. Arrangement as claimed in claim 22,  
characterized in that a proxy represents one and only one object, a proxy being deleted when the represented object terminates, alternatively that an object can be represented by multiple proxies, e.g. one proxy for each client object.

24. Arrangement as claimed in any of the claims 1-3 or 18-23,  
characterized in that any object wanting to interact with objects in a different domain are registered in the mobility functions (M), the registration of the necessary objects for an application being done for example at the initialisation by an object which is customised for the configuration, the definition and creation of a proxy conveniently being done together with the registration of the object it is representing.

25. Arrangement as claimed in claim 24,  
characterized in that only the grouping of objects into clusters, capsules and nodes is performed, all objects, clusters, capsules and nodes being considered as if they are all located within the same domain, the application designer adding thereto specific programming code and executing the linking and compilation.

26. Arrangement as claimed in any of the claims 1-3 or 19-25,  
characterized in that there is introduced a new redirection function on the DPE in question,

as well as the possibility of generating special stub for any dynamic objects.

add  
B1

~~add~~  
C' )